UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/804,033	03/19/2004	Jim A. McAlear	42004-0003	7065

41618 7590 12/19/2008

CASSAN MACLEAN
307 GILMOUR STREET
OTTAWA, ON K2P 0P7
CANADA

EXAMINER	
VU, TUAN A	
ART UNIT	PAPER NUMBER
2193	
MAIL DATE	DELIVERY MODE
12/19/2008	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary	Application No.	Applicant(s)	
	10/804,033	MCALEAR, JIM A.	
	Examiner	Art Unit	
	TUAN A. VU	2193	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *03 October 2008*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *19-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *19-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application
6) ☐ Other: _____.

DETAILED ACTION

1. This action is responsive to the Applicant's response filed 10/3/08.

As indicated in Applicant's response, claims 1-18 have been canceled. Claims 19-37 are pending in the office action.

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

3. Claims 19-24, 26-31 are rejected under 35 U.S.C. 102(b) as being anticipated by Moorby et al, USPN: 5,892,507 (hereinafter Moorby).

As per claim 19, Moorby discloses a method of illustrating a software process in a diagram, the software process including software objects and operations (e.g. animation audio sequence, OLE object, Call … Subroutine - col. 8 lines 11-25), each operation being associated with a respective one of the software objects, and control flow mechanisms determining a control flow of the software process (SVL – col. 3 lines 24 to col. 4 line 5; SVL objects – col. 11, lines 1-4),

each operation being associated with a respective one of the control flow mechanisms, the diagram being characterized by a horizontal direction and a vertical direction (Fig. 1; Fig. 14-16; Fig. 20, 25a, 26, 28), the method comprising:

placing in the diagram, for each control flow mechanism, a control flow segment (icon 1 icon 2 icon 3 – Fig. 14) representing the control flow mechanism, and further

placing the control flow segments in the diagram in series so as to form a timeline extending longitudinally in the horizontal direction (*timelines* - Fig 11a-c and related text at col. 11), the timeline illustrating the control flow of the software process including a sequence of the operations (col. 3, lines 60 to col. 4, line 15; storyline 200 – Fig. 9; Fig. 14-16 -- Note: *Caller Spot, Start Spot, Go-to, Break, End Spots, Triggerable Icon Button ... clicking* - col. 7 line 10 to col. 8 line 56 – reads on operations sequenced along the timelines segment, each segment with a start point -- see 402, 434, 450, Fig 14-16; Fig 25a,b), each control flow segment defining at least one respective vertical column (icon 1, 414 line; icon 2, line 416; icon 3, line 418 – Fig. 14);

placing in the diagram, for each software object, an elongated shape representing the software object, the elongated shape extending longitudinally in the horizontal direction, the elongated shape containing text specifying the software object (Fig. 11c; Fig. 12A-B; segment 454: 462 segment 458: 468, Fig 16), and further

placing each elongated shape in the diagram so as to be parallel to the timeline (Fig. 11C, Fig. 12B) and spaced apart in the vertical direction from the timeline;

placing in the diagram, for each operation, a compact shape representing the operation, the compact shape identifying an operation type of the operation (e.g. col. 8 lines 11-25; Fig. 25a,b), the compact shape containing a symbol particularly specifying the operation (e.g. icon 1, 414 line; icon 2, line 416; icon 3, line 418 – Fig. 14; Fig. 17-19), and further

placing each compact shape in the diagram so as to be adjacent the elongated shape representing the software object associated with that operation (Fig. 14-20), thereby illustrating the association of that operation with that software object, and further

placing the compact shape in the vertical column (icon 1, 414 line; icon 2, line 416; icon 3, line 418 – Fig. 14; Fig. 17-19 – Note: vertical lines joining a iconic action read on vertical column) defined by the control flow segment representing the control flow mechanism associated with that operation (col. 8 lines 11-25; Fig. 25a,b), thereby also illustrating the association of that control flow mechanism with that operation;

whereby the software process (e.g. *underlying program processes… call spots, path line, action icons, links… sequence to be played* - col. 2 line 53 to col. 4 line 52) is illustrated in the diagram.

As per claim 20, Moorby discloses wherein a first one of the operations is associated with a first one of the software objects, and a second one of the operations is associated with a second one of the software objects, (Fig. 25a-b; *timelines* - Fig 11a-c and related text, storyline 200 – Fig. 9; Fig. 14-16) the first operation producing data based on the first software object, the second operation operating on the second software object based on the data (e.g. Fig. 25a,b; Fig. 27), the method further comprising:

further placing in the diagram first and second ones of the elongated shapes respectively representing the first and second software objects (e.g. Fig. 9; Fig. 14-16 -- Note: *Caller Spot, Start Spot, Go-to, Break, End Spots, Triggerable Icon Button … clicking* - col. 7 line 10 to col. 8 line 56 – reads on operations sequenced along first and second timelines elongated segments, each segment with a start point -- see 402, 434, 450, Fig 14-16; Fig 25a,b), such that a respective part of each of those elongated shapes is contained in a particular one of the vertical columns, and such that those elongated shapes are spaced apart in the vertical direction (e.g. icon 1, 414 line; icon 2, line 416; icon 3, line 418 – Fig. 14; Fig. 17-19);

further placing in the diagram, in the vertical column(e.g. icon 1, 414 line; icon 2, line 416; icon 3, line 418 – Fig. 14) containing the respective parts of the first and second elongated shapes, first and second ones of the compact shapes respectively representing the first and second operations (Fig. 14-20), the first compact shape identifying the operation type of the first operation as producing the data from the first software object, the second compact shape identifying the operation type of the second operation as operating on the second software object (e.g. *setZ, isZeq19* , Fig. 25a; *isZeq17, isZlt18, Z = Z + 1* , Fig. 25b); and

further placing in the diagram a line connecting the first and second compact shapes thereby illustrating that the operation of the second operation on the second software object is based on the data produced by the first operation based on the first software object (e.g. Fig. 14-20; col. 8 lines 11-25; Go to # 1 → Go back – Fig. 27).

As per claim 21, Moorby discloses wherein the first and second compact shapes are spaced apart in the horizontal direction thereby illustrating that the second operation follows the first operation in the sequence of the operations (refer to Fig. 25a,b; *underlying program processes… call spots, path line, action icons, links… sequence to be played* - col. 2 line 53 to col. 4 line 52).

As per claim 22, Moorby discloses wherein the first compact shape is spaced apart in the vertical direction from the first elongated shape thereby illustrating that the first operation does not modify the first software object (*Go to 918, 920* icon – Fig. 27; no blocking *step 808* Fig. 25 – Note: Go to branching in vertical line and icon reads on spaced from elongated time segment 916 with no modification of object; and non blocking 808 spots reads on not affecting data of a timeline segment), and

wherein the second compact shape touches the second elongated shape thereby illustrating that the second operation modifies the second software object (e.g. setZ, IncrZ Fig. 25 - Note: *nonblocking spots* – see Fig. 5 -- allowing the other threads to operate **reads on** not touching the timeline and/or changing any object therein, while blocking *setZ, incrZ* call spots – see Fig. 5 **-- reads on** touching the timeline segment data).

As per claim 23, Moorby discloses wherein a third one of the operations is associated with a third one of the software objects, the third operation producing data based on the third software object, and wherein the control flow mechanism associated with the third operation is conditional upon the data produced by the third operation (Fig. 25a,b - Note: segments in timeline axis and iconic call spots reads on 1st, 2nd 3rd operation whose respective control flow mechanism is conditional upon data produced therein), the method further comprising:

further placing in the diagram a line connecting the compact shape representing the third operation and the control flow segment which represents the control flow mechanism associated with the third operation, that compact shape identifying the operation type of the third operation as producing the data from the third software object (Fig. 25, 27 – refer to claim 20);

thereby illustrating that the control flow mechanism represented by that control flow segment is conditional upon the data produced by the third operation based on the third software object (Fig. 14-20; col. 8 lines 11-25; Go to # 1 → Go back – Fig. 27 - refer to claim 20).

As per claim 24, Moorby discloses wherein fourth and fifth ones of the software objects are related, the method further comprising:

further placing in the diagram fourth and fifth ones of the elongated shapes respectively representing the fourth and fifth ones of the software objects, such that a respective part of each

of those elongated shapes is contained in a particular one of the vertical columns, and such that those elongated shapes are spaced apart in the vertical direction (Fig. 14-20; refer to claim 19);

further placing in the diagram a relationship symbol adjacent the fourth elongated shape, and further connecting the relationship symbol to the fifth elongated shapes with respective lines, thereby illustrating the relationship of the fourth and fifth software objects (e.g. setZ, isZeq19 - Fig. 25a-b).

As per claim 26, Moorby discloses wherein a particular one of the software objects is associated with a plurality of the operations, that plurality of operations being associated with a respective plurality of the control flow mechanisms, whereby the control flow segments respectively representing such plurality of control flow mechanisms define a respective plurality of vertical columns (refer to claim 19), and wherein each one of such plurality of vertical columns contains at least one of the compact shapes respectively representing that plurality of operations, and wherein each one of such plurality of columns further contains a respective portion of the elongated shape representing that software object (segment 592 - Fig. 20; segments 736, 738 – Fig. 24; *eventspace* 852 containing segments *866, 870, 872* - Fig. 26).

As per claim 27, Moorby discloses wherein the software process further includes a list assignment or a parameter specification (OMF 3 612 – Fig. 21), the method further comprising placing in the diagram a further shape containing details of the list assignment or the parameter specification (OMF1 Fig. 4a; OMF 612 Fig. 612), respectively.

As per claim 28, Moorby discloses wherein the software process further includes a mathematical expression, the method further comprising placing in the diagram an even further shape containing the mathematical expression (e.g. shape 816, 814, 822 Fig. 25B).

As per claim 29, Moorby discloses wherein at least one of the control flow mechanisms is selected from a group comprising: looping; nested looping; conditional branching nested branching (Go to 918, 920, go back 922 Fig. 27; Recycle Fig. 2F; Fig. 2C, 2d, Fig. 2B).

As per claim 30, Moorby discloses wherein the text contained by at least one of the elongated shapes (see Fig. 11c Note: elongated shapes represented by Icon reads on text contained therein) specifies that the software object represented by that elongated shape includes: an array (BMAP 1, Fig. 4a); a table; a file (Meta1, script 1, BMAP 1 – Fig. 4a); a queue (SEQ 1 Fig 4B; SEQ7 Fig. 13b); a tree structure (wksp 1 60, Fig. 4b); or a software variable (Fn1 Fig. 4D; Fig. 25a,b).

As per claim 31, Moorby discloses wherein the symbol contained by at least one of the compact shapes specifies the operation represented by that compact shape as including: selecting(Fig. 2d, 2f); substitution (action 956 – Fig. 28); formatting; copying (edited - col. 9 line 66 to col. 10 line 7; action 956 – Fig. 28); making an assignment (setZ - Fig. 25); making a state change (col. 9 lines 25-29; Fig. 2a,b,c,d, e; Fig. 12a; action 956 – Fig. 28); making a computation or returning a value (Fig. 25a,b).

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

5. Claim 25 is rejected under 35 U.S.C. 103(a) as being unpatentable over Moorby et al, USPN: 5,892,507, and further in view of Charisius et al., USPN: 7,055,131 (hereinafter Charisius).

Moorby discloses wherein the fifth software object is related to the fourth software object by data content (Global Z – Fig. 25), or interface (Function IsZeq, Sub setZ; call spot/icon 590, icon 594, interface 592 – Fig. 20; interface b/w 816 and 822 - col. 16 lines 47-65); but does not disclose relationship being inheritance or encapsulation. However, Moorby discloses encapsulation of objects (OLE 50 Fig. 4a) in a design framework, reuse of objects in an authoring tool including library for the author to select objects from (see col. 4, line 67 to col. 5, line 5; *library* – col. 6, line 65 to col. 7, line 5), a meta file and OMF object having therein representation of meta information interrelating objects of a framework (see Meta 588, OMF 612, Fig. 20-21). This is reminiscent of in software development wherein a repository (database or library) of reusable objects can be interrelated via definitions of a form of meta-information (Meta file) that enable rebuilding and interlinking (as suggested by Moorby: (e.g. *linking* - col. 10 ,lines 11-29) these database objects such as re-structuring or modeling during authoring runtime a framework of objects (OMF) as set forth above. **Charisius** discloses in an authoring system for development of object-oriented type of software process, using meta information (TMM – Fig. 2) to interrelate stored package of objects or class in Microsoft template library (e.g. Fig. 9; Fig. 4) consistent with the concept of Object-oriented reusable package (see Polymorphism - Table 8, col. 12) using UML and Rationale Rose (UML – col. 18, lines 25-32); and similarly to Moorby discloses a tool to animate a program flow/process via a time-based GUI representation of event-icon acting via links upon a linear sequential representation of the

objects flow (see Fig. 13-14, 21). Based on standard visual programming known to include reusable library or repository of objects at the time the invention was made, it would have been obvious for one skill in the art to implement the meta file and the OMF by Moorby so the interrelationship (or links) among objects therein represent relationship of OO objects consistent with inheritance and object encapsulation (as in UML or Rationale Rose), definitions (as by meta-information) and database table relationships (as queried from any retrieval of OO package or source repository of reusable OO objects) as evidenced above by Charisius. One would be motivated to do so because at the time the invention was made, reusability of objects being stored and queried from a repository based on a meta information within an instance of a framework type for software construction was a well known concept as mentioned by Charisius (see Fig. 1), and using repository of OO packages in conjunction with metadata as shown in Moorby's OMF and Metafile would enable modeling and framework support for interrelating objects used within a process animation endeavor such as Moorby including benefits from their inheritance and how they are re-instantiated and validated based on their regulated database store (as in RDBMs) and hierarchy as evidenced in Charisius' use of persisted objects assembled for a instance model (see Fig. 10-26).

6. Claims 32-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Moorby et al, USPN: 5,892,507, and further in view of Garmon et al, USPN: 5,355,450 (hereinafter Garmon).

As per claim 32, Moorby discloses a graphical user interface of a computer aided design software tool in a computer system for illustrating a software process according to the method of claim 19, the graphical user interface comprising:

a grid for the placement of the control flow segments, each of the compact shapes, and each of the elongated shapes(Fig 9, 11 – Note: elongated timeline segment interlaced with vertical lines representing action acting upon a segment reads on grid), in the diagram;

a first set of activatable controls (col. 2 line 23 to col 3 line 24 -- Note: graphical tool enabling elongated shapes to be displayed and user-manipulated reads on activatable controls pertinent to tool) for placing each of the control flow segments on the grid;

a second set of activatable controls for placing each of the compact shapes on the grid (refer to Fig. 14, 17, 18, 20, 26 – Note: graphical tool where icons are user manipulated reads on activatable controls for placing icons on grid), the graphical user interface presenting to the user a dialog box for entry of the symbol specifying the operation represented by that compact shape when that control is activated; and

a third activatable control for placing each of the elongated shapes on the grid (Fig. 14, 17, 18, 20, 26 – see claim 19),

Moorby does not explicitly disclose graphical user interface presenting to the user a dialog box for entry of the text specifying the software object represented by the elongated shape when that control is activated. However, graphical tool enabling the user to edit content such that the content can be textual is evidenced in Moorby (col. 10 lines 40-59) and video editing based on *Avid Technology* (col. 11 lines 32-44) known tool for composing a storyline. Garmon, in said Avid Technology, discloses a graphical tool to specify parameters for video segments analogous to Moorby, discloses dialog box (Garmon: Fig. 3) for entering textual specifications as to how actions would be performed. It would have been obvious for one skill in the art at the time the invention was made to implement Moorby's video and textual editing so that as in Avid

Technology, dialog boxes are provided for entry of the text specifying the software object represented by the elongated shape when that control is activated, because to active a control, some parametric inputs can be modified according to the developer as endeavored in Moorby, and using dialog box as in Garmon would help facilitate instant modification or re-specification of how actions can be adjusted (as in Avid Technology) for the intended video story telling as set forth above.

As per claim 33, Moorby discloses wherein at least one of the control flow mechanisms is selected from a group comprising: looping; conditional branching; nested branching; exception branching; and thread handling (refer to claim 29), and the first set of activatable controls includes a specific activatable control for that control flow mechanism (refer to claim 32).

As per claim 34, Moorby (in view of the obviousness rationale in claim 32) discloses wherein the dialog box for entry of the text specifying the software object is for entry of the text specifying that the software object includes: an array; a table; a file; a queue; a tree structure; or a software variable (refer to claim 30).

As per claim 35, Moorby (in view of Garmon) discloses wherein the operation type of at least one of the operations includes: selecting; substitution; formatting; copying; making an assignment; making a state change; making a computation or returning a value (refer to claim 31) and the second set of activatable controls (refer to claim 32) includes a further specific activatable control for that operation type.

As per claim 36-37, Moorby discloses a computer program product for use with the computer system, the computer program product comprising a tangible computer-readable medium having encoded thereon computer-readable code for implementing the graphical user

interface according to claim 32 (e.g. Fig. 8) wherein the tangible computer-readable medium comprises a diskette, a CD-ROM, a fixed disk, or a memory device including a semiconductor, magnetic, or optical memory device (e.g. Fig. 8).

Response to Arguments

7. Applicant's arguments filed 10/03/08 have been fully considered but they are mostly moot in light of the new grounds of rejection which have been necessitated by the Amendments.

Regarding the argument that Moorby does not teach 'operation on the software object and control flow mechanism ... are contained in a single vertical column ... being perpendicular to the horizontally ... timeline' (Appl. Rmrks pg. 12 middle), it is observed that the above teaching is nowhere to be found in such literal form in the claim language. The argument is deemed not commensurate with the specifics of the claimed subject matter, hence not convincing. Applicant's arguments fail to comply with 37 CFR 1.111(b) because they amount to a general allegation that the claims define a patentable invention without specifically pointing out how the language of the claims patentably distinguishes them from the reference.

Conclusion

8. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Tuan A Vu whose telephone number is (571) 272-3735. The examiner can normally be reached on 8AM-4:30PM/Mon-Fri.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Lewis Bullock can be reached on (571)272-3759.

The fax phone number for the organization where this application or proceeding is assigned is (571) 273-3735 (for non-official correspondence - please consult Examiner before

using) or 571-273-8300 (for official correspondence) or redirected to customer service at 571-272-3609.

Any inquiry of a general nature or relating to the status of this application should be directed to the TC 2100 Group receptionist: 571-272-2100.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

/Tuan A Vu/

Primary Examiner, Art Unit 2193

December 16, 2008